LITHAN

EDU CLaaS® DIGITAL ACCELERATOR

**Module:** **Front End Web Development**

# IU 03. Apache & FTP

By the end of this session, you will be able to understand Apache Web Server & FTP Tools

# Instructional Units in the Module

| IU # | IU Description | Required / Optional |
|---|---|---|
| 01 | Website Technologies & Tools | Required |
| 02 | Website Design Overview | Required |
| **03** | **Apache & FTP Tools** | **Required** |
| 04 | Introduction to HTML & Web Browsers | Required |
| 05 | Structuring & Styling with CSS | Required |
| 06 | Working with CSS : An Example | Required |
| 07 | Introduction to Javascript | Required |
| 08 | Introduction to Jquery & JQueryUI | Required |
| 09 | HTML 5 and CSS 3 | Required |
| 10 | Testing Interactive websites & Standards | Required |

CLaaS®

# IU Content

| S. No. | Topic Description | Required / Optional |
|---|---|---|
| 01 | Install Textpad | Required |
| 02 | Install Win Merge | Required |
| 03 | Install Apache Web Server | Required |
| 04 | What is a Web Server ? | Required |
| 05 | Introduction to Apache Web Server | Required |
| 06 | Configuring Apache Web Server | Required |
| 07 | Virtual Hosts | Required |
| 08 | Monitoring Apache Logs | Required |
| 09 | Apache & Common Modules | Required |
| 10 | Apache File Access Control | Required |
| 11 | Securing a Folder | Required |
| 12 | Filezilla FTP Server Setup | Required |

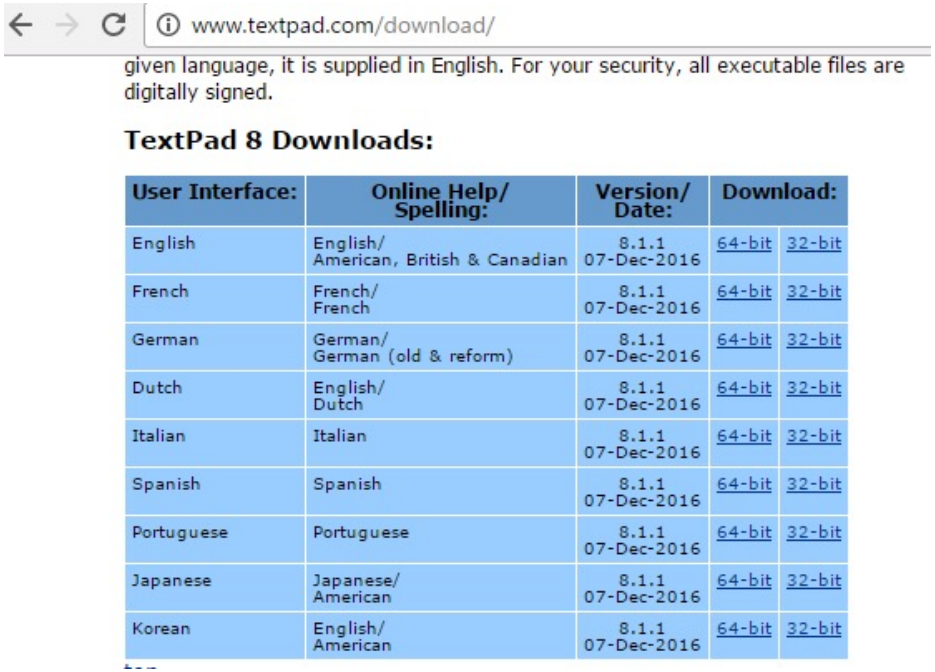
# IU Content

| S. No. | Topic Description | Required / Optional |
|---|---|---|
| 13 | Configuring FileZilla | Required |
| 14 | Connecting to FTP Server & Do Upload & Download | Required |
| 15 | Backup & Restore Files | Required |

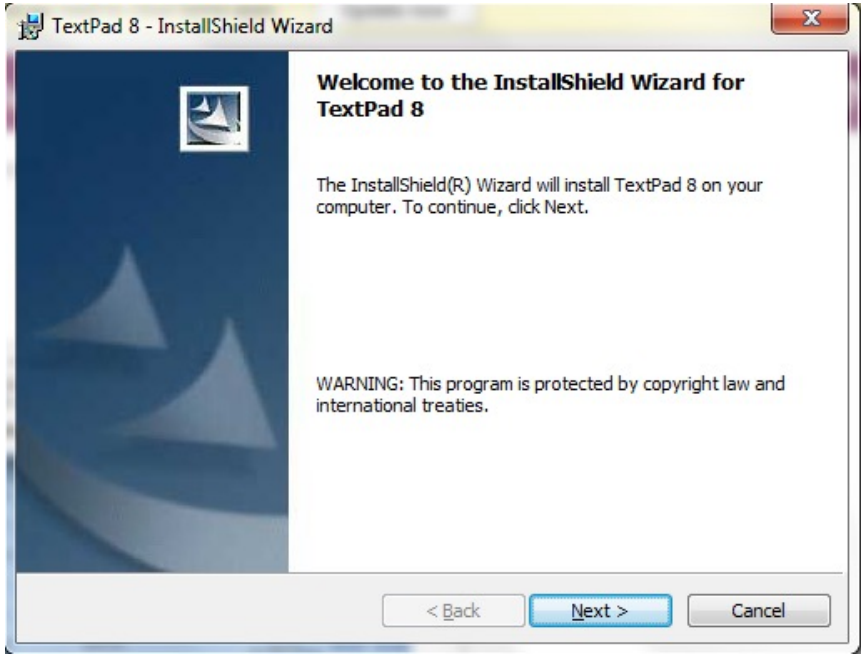
CLaaS®

# Install Textpad

- Download Textpad 8 from http://www.textpad.com/download/
- Unzip the Package and Click Setup.exe
- Follow the instructions to Setup TextPad 8

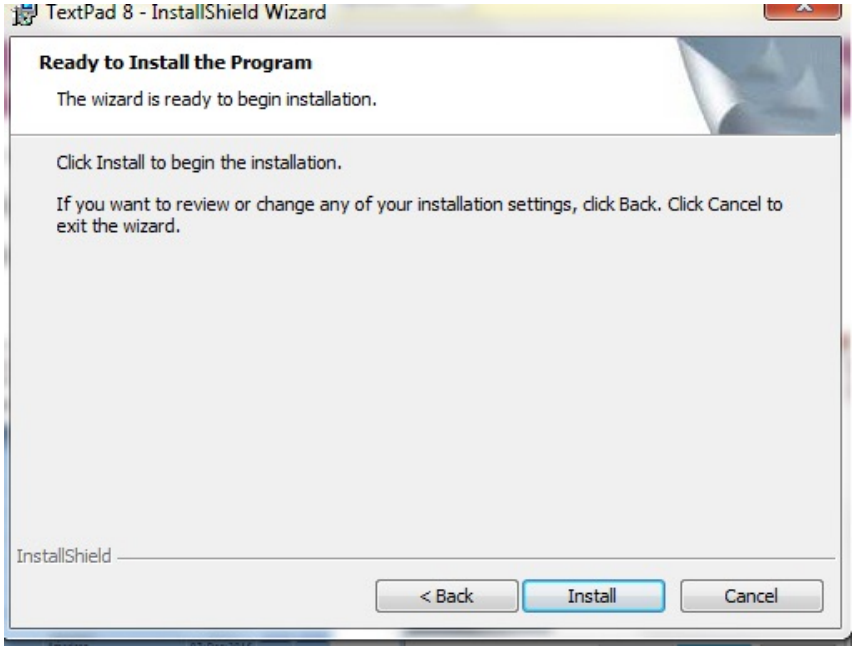


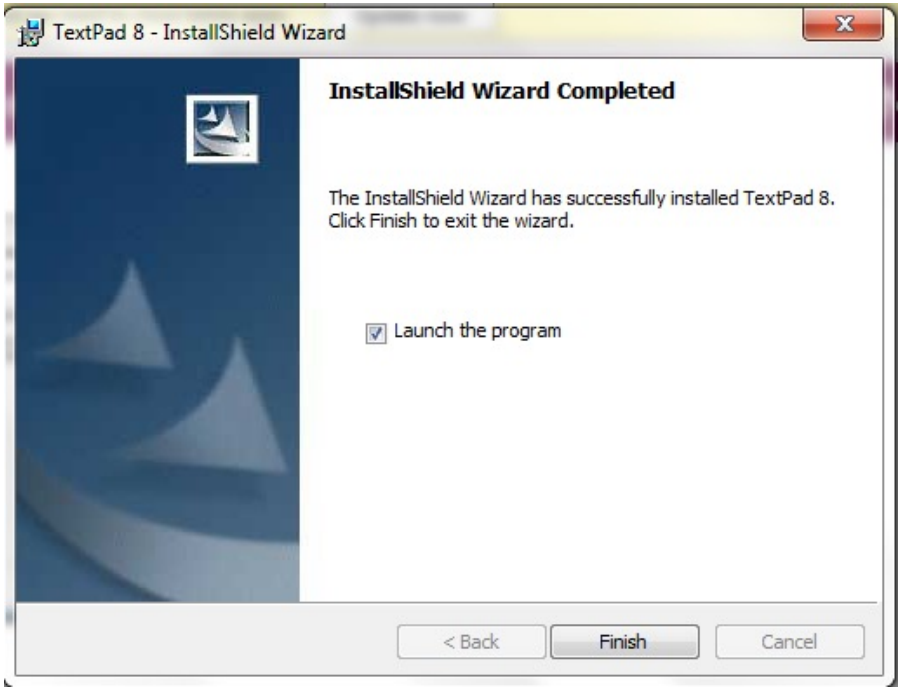

| User Interface: | Online Help/ Spelling: | Version/ Date: | Download: | |
|---|---|---|---|---|
| English | English/ American, British & Canadian | 8.1.1 07-Dec-2016 | 64-bit | 32-bit |
| French | French/ French | 8.1.1 07-Dec-2016 | 64-bit | 32-bit |
| German | German/ German (old & reform) | 8.1.1 07-Dec-2016 | 64-bit | 32-bit |
| Dutch | English/ Dutch | 8.1.1 07-Dec-2016 | 64-bit | 32-bit |
| Italian | Italian | 8.1.1 07-Dec-2016 | 64-bit | 32-bit |
| Spanish | Spanish | 8.1.1 07-Dec-2016 | 64-bit | 32-bit |
| Portuguese | Portuguese | 8.1.1 07-Dec-2016 | 64-bit | 32-bit |
| Japanese | Japanese/ American | 8.1.1 07-Dec-2016 | 64-bit | 32-bit |
| Korean | English/ American | 8.1.1 07-Dec-2016 | 64-bit | 32-bit |




CLaaS®

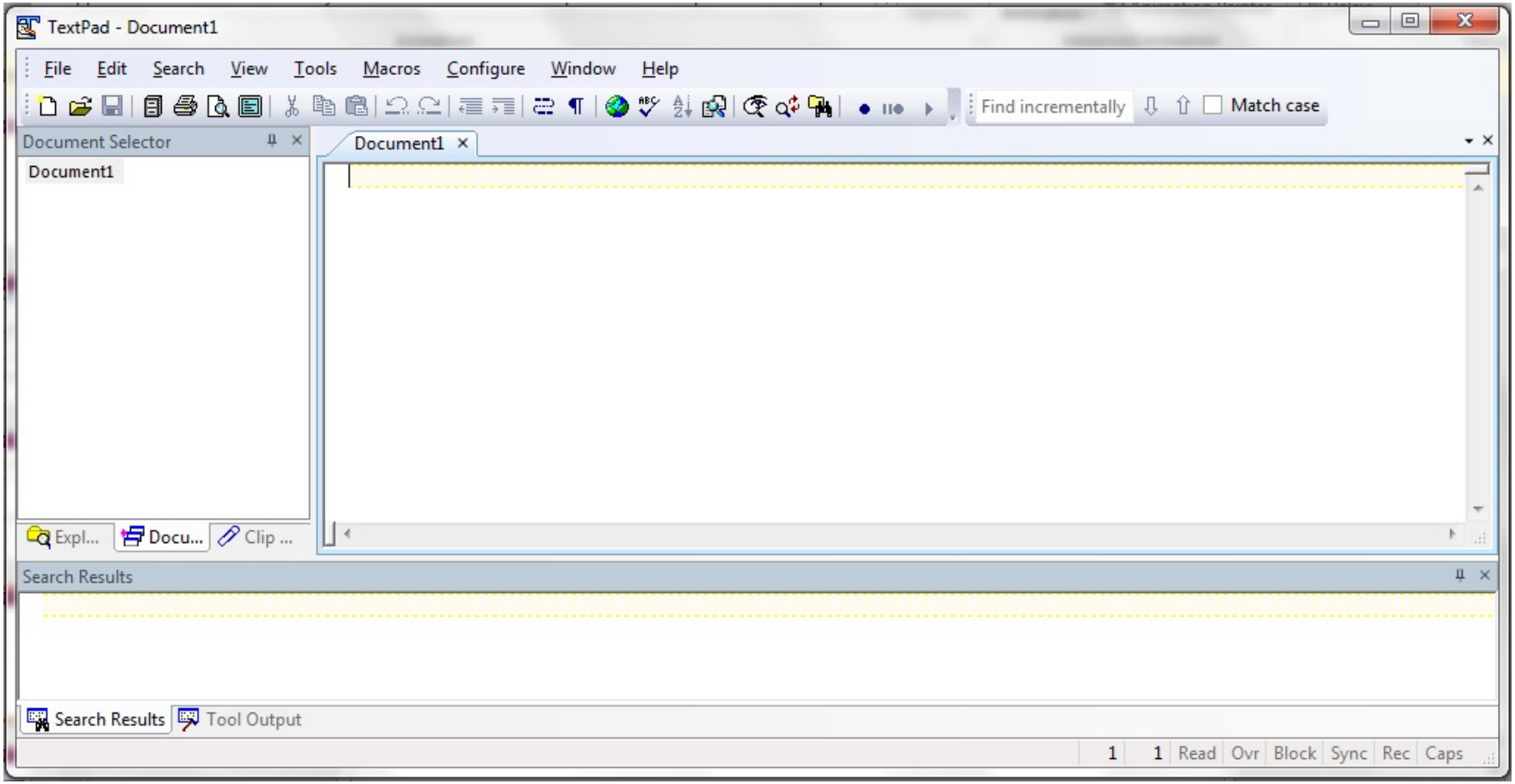
# Follow the Steps & Install Textpad




CLaaS®

# Launch Textpad & Verify




CLaaS®

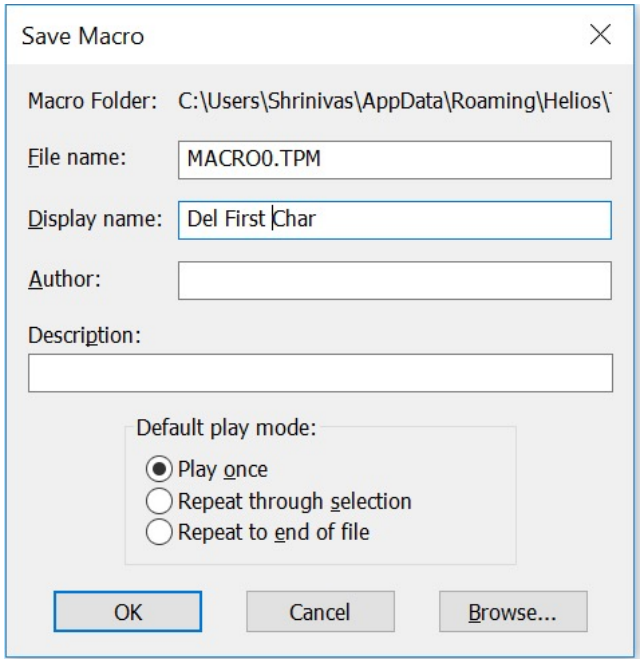
# Configuring Syntax Highlighting in Textpad

- TextPad can highlight syntax for Various Programming Languages
- Download Syntax files from http://www.textpad.com/add-ons/syna2g.html
  - Download for Python & PHP (Refer to Notes Below)
  - The files will have the extension .syn
  - Place the files in C:\Program Files\TextPad 8\system
- Open TextPad & Go to menu Configure -> New Document Class
- Type in PHP & Click Next
- Type in *.php & Click Next
- Select the Checkbox Enable Syntax Highlighting
- Choose the Syn file downloaded above & Click Next
- Click Finish
- Do the same for *.py
- Now if you open *.py file or *.php file, it will highlight Syntax

CLaaS®

# Textpad Record Macro

- Textpad allows you to record Key sequences and replay it up to the end of file.
- Example, if you want to delete 1st letter in every line you do the following
  - Position the Cursor in 1st Line
  - Click Macros -> Record
  - Click Home Key
  - Click delete key
  - Click Down Arrow
  - Click Macro -> Stop Recording
  - Save the Macro in a File
  - Give the Display Name “Del First Char”
  - Click OK



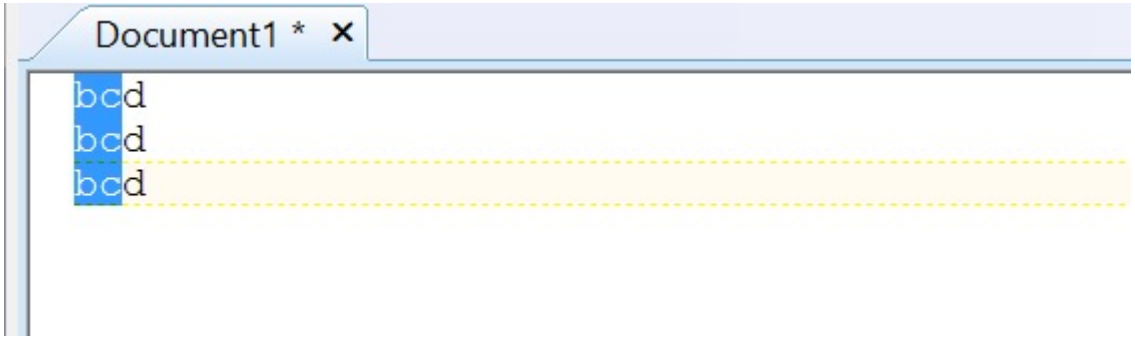

- To replay it up to end of file by clicking Macro-> Multi Play & Select “Repeat to End of File”, choose the Macro and click Ok
- The 1st Char in all line will be deleted

CLaaS®

# Other Textpad Features

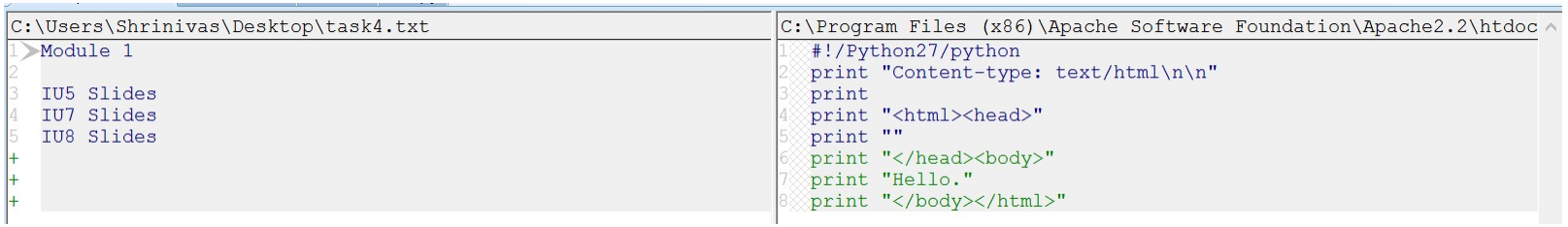
## ❑ Block Select Mode

- Allows you to select a Rectangular area in the text and delete them
- Right Click the Main Window and Choose Block Select Mode, Now you can select, 1st 2 letters in each line




## ❑ File Comparison

- You can choose Tools -> Compare Files to compare 2 files

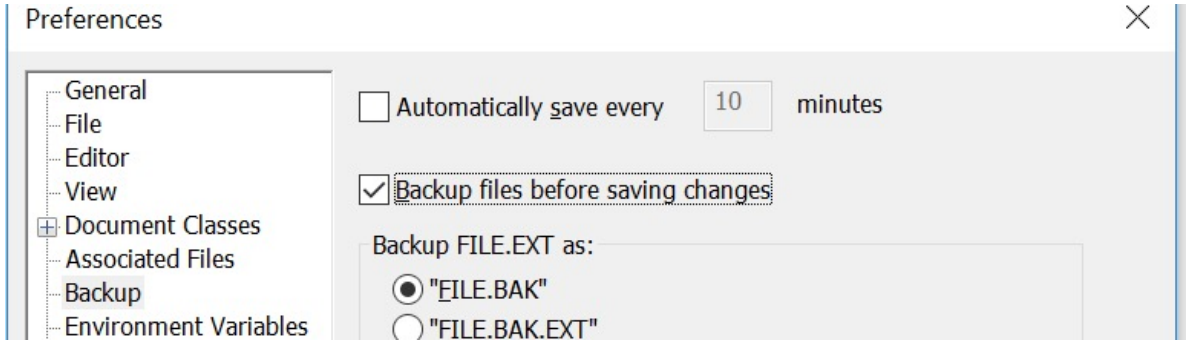



CLaaS®

# Other Textpad Features

## ❑ Backup Files before Modification

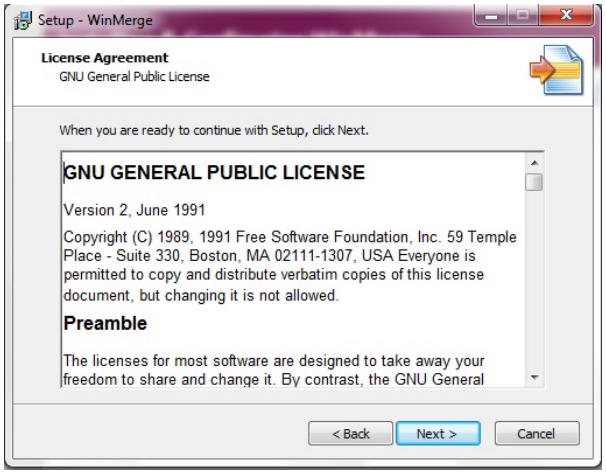
- Click Configure -> Preferences -> Backup
- Check Backup File before saving

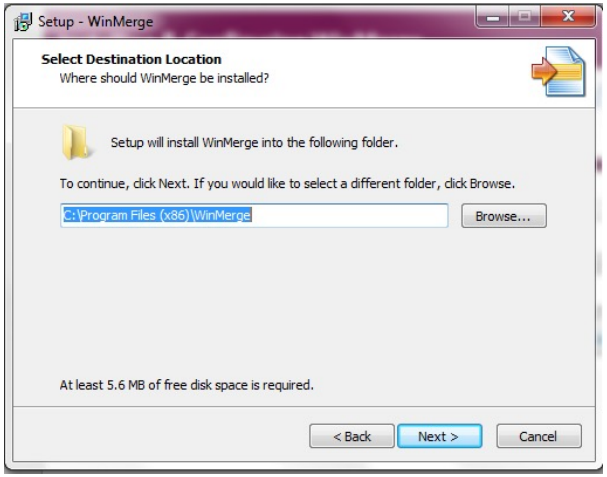


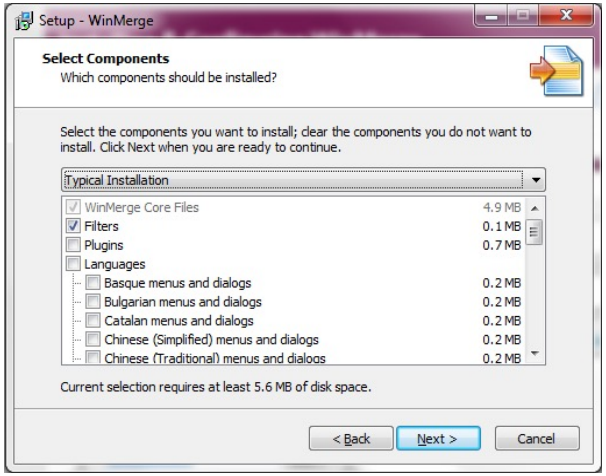

CLaaS®

# Installing & Configuring WinMerge

- Download WinMerge from http://winmerge.org/?lang=en
- Open the Installer

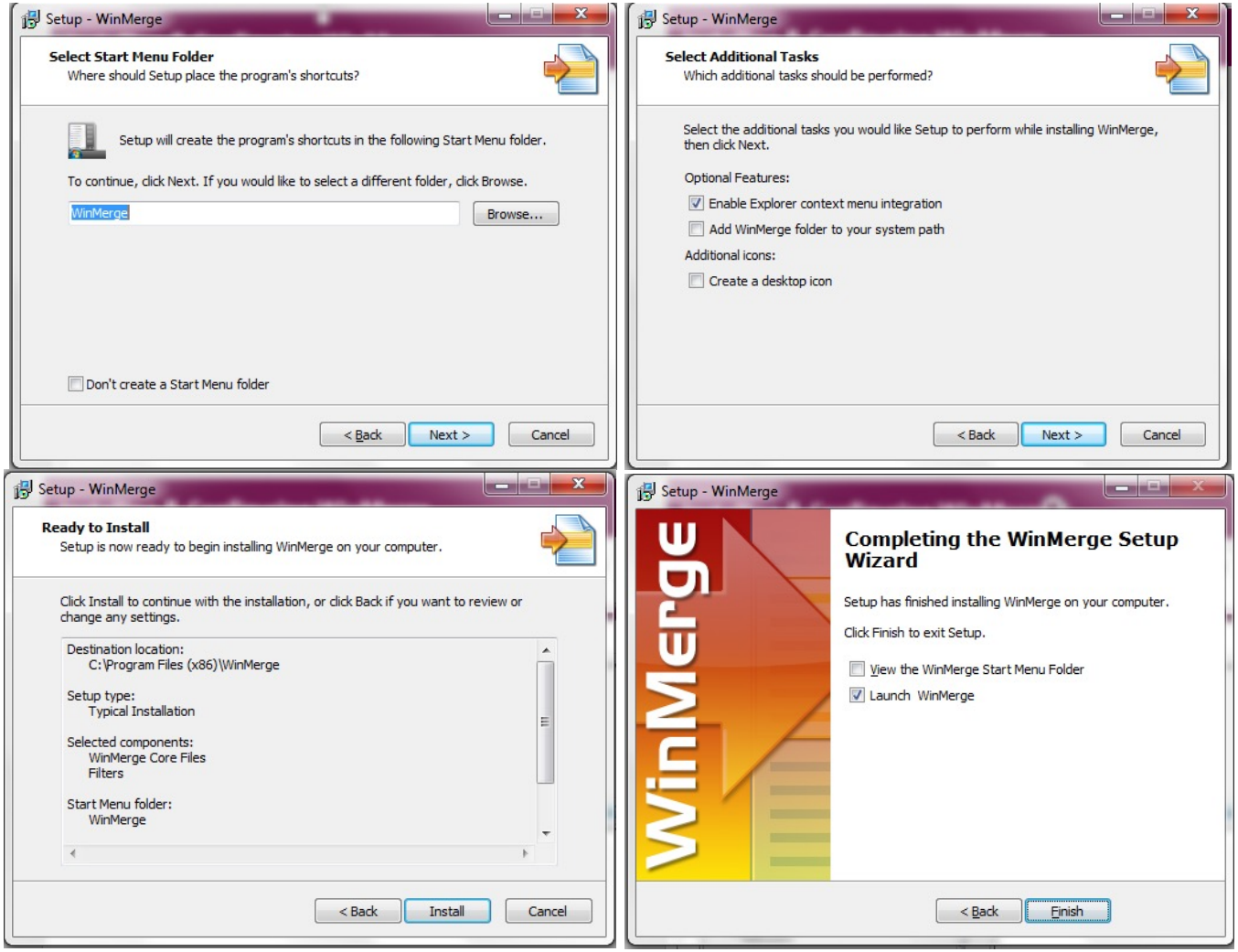




CLaaS®

# Installing & Configuring WinMerge

- WinMerge is used for Comparing Files & Folders



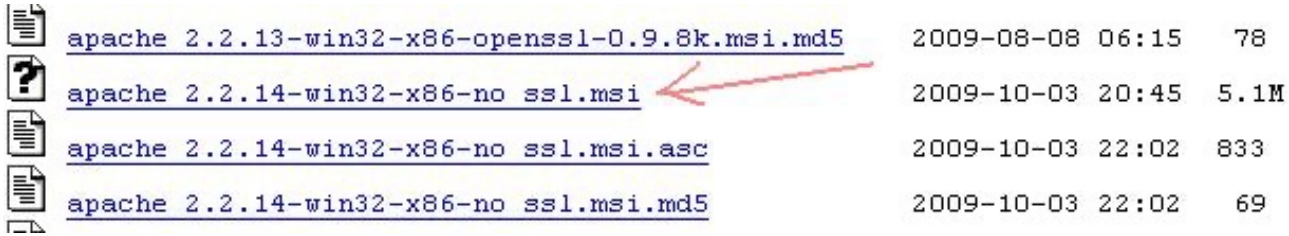

CLaaS®

# Features of WinMerge

- WinMerge is an Open Source differencing and merging tool for Windows.
- WinMerge can compare both folders and files, presenting differences in a visual text format that is easy to understand and handle.
- WinMerge can Synchronize folders by deleting, updating or adding files from source to target

# Installing Apache Web Server

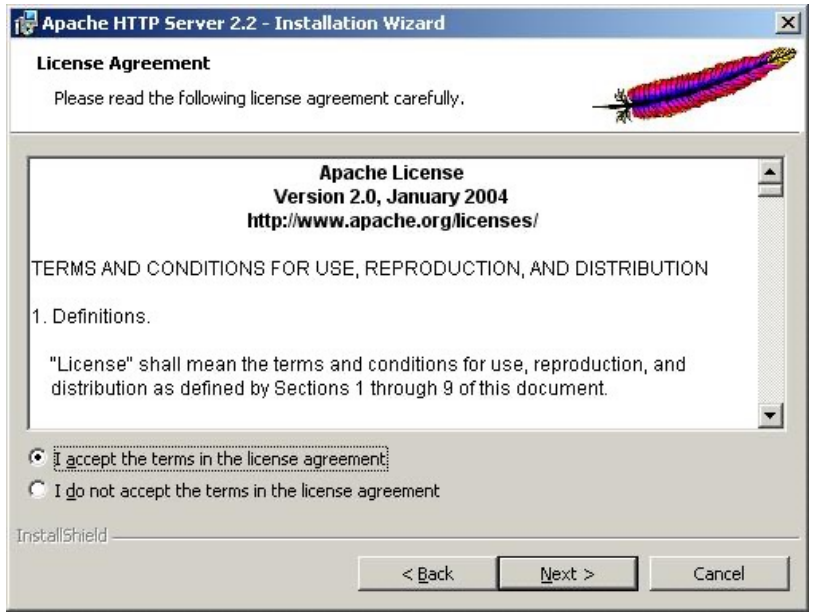
- **Step 1: Download Apache 2.2**
  - We use Apache 2.2.14 to demonstrate the installation process. Other 2.x versions have very similar installation steps.
  - Go to Archived Win32 binaries download page here http://archive.apache.org/dist/httpd/binaries/win32/
  - Download Apache 2.2.14 msi installer.
  - Here is the direct download link apache_2.2.14-win32-x86-no_ssl.msi http://archive.apache.org/dist/httpd/binaries/win32/apache_2.2.14-win32-x86-no_ssl.msi

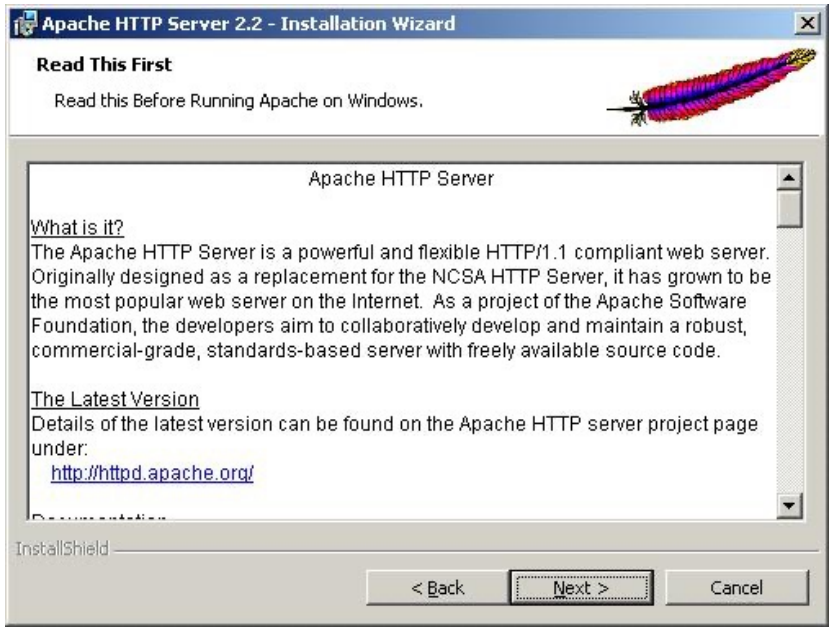



- **Step 2: Start installation**
  - Double click the installation file. The installation should start straight away. Here are 9 screenshots.

CLaaS®

# Installing Apache Web Server

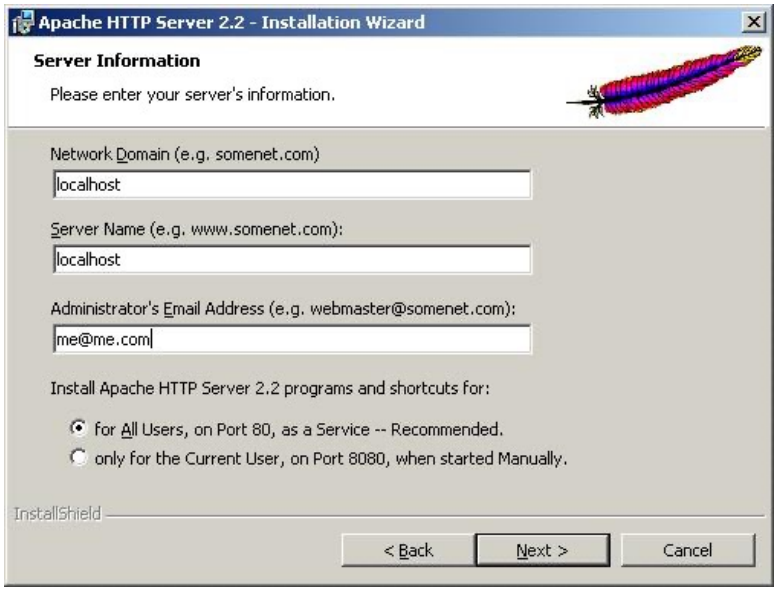



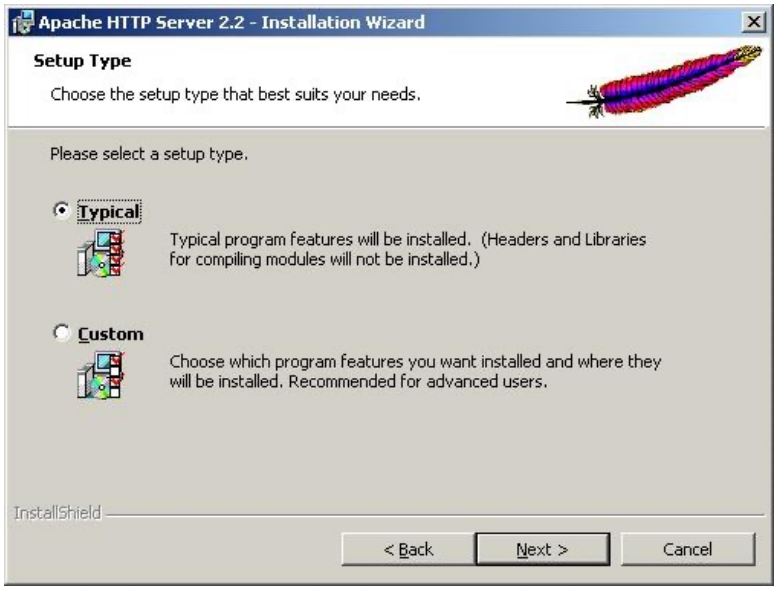




CLaaS®

# Installing Apache Web Server

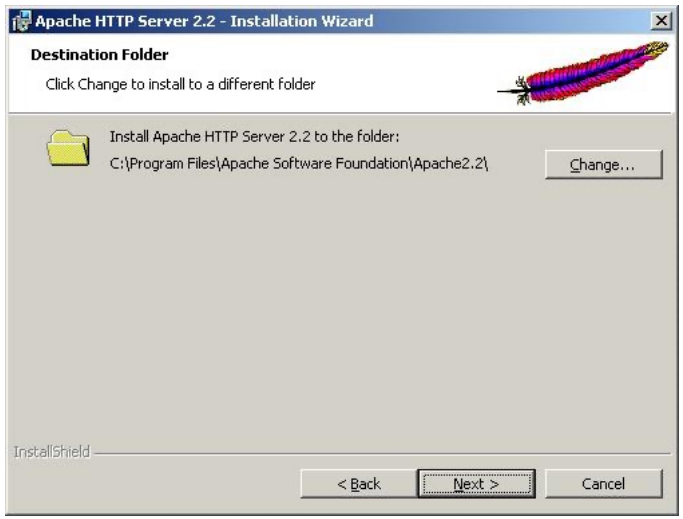
- This step needs a bit explanation. Add localhost to both Network Domain and Server Name box. Administrator's Email Address can be anything you like. These info are used for Apache configuration file but you can change them after the installation.
- Next, select the first radio button as recommended. This way, Apache will be running as a Windows Service. See Step 6 below for more info about how to start, stop, restart Apache.

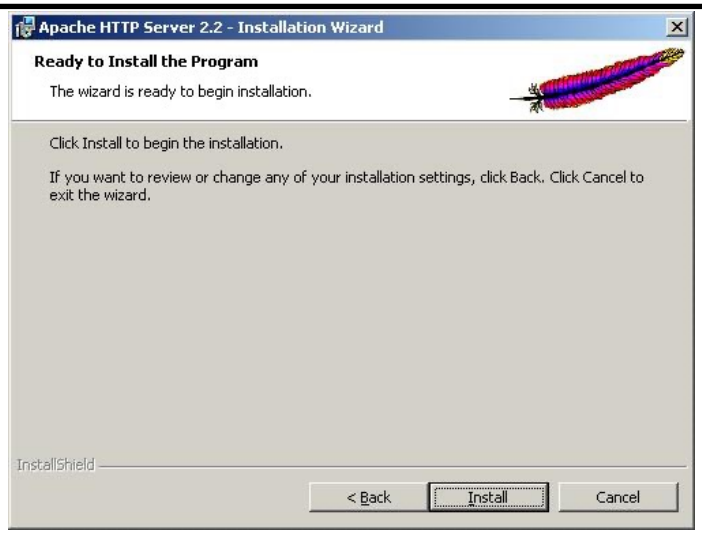


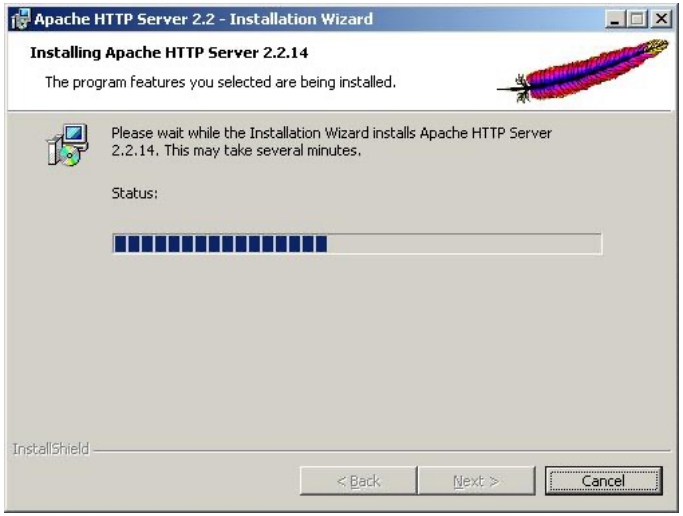



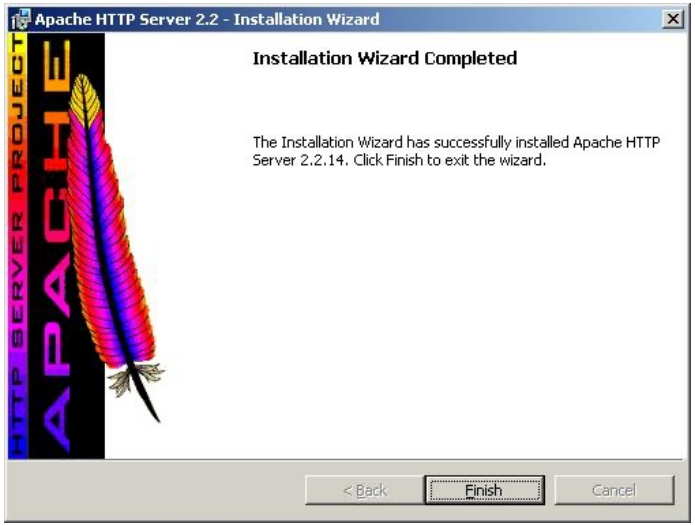

CLaaS®

# Installing Apache Web Server



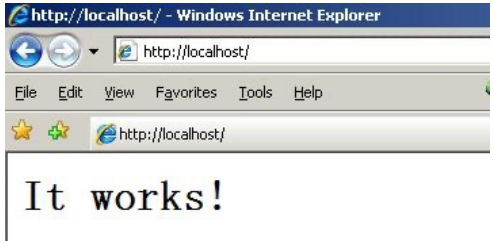




CLaaS®

# Test Your Apache Web Server Installation

- After install, open web browser and type either of the following URLs into your browser's address bar and hit Enter key.
  - http://localhost
  - http://127.0.0.1
  - http://169.254.32.57 or 192.168.0.1 (eg. local network IP address)
- If Apache has been installed correctly, you should see a success message opened in your web browser, as shown below.




- The text actually comes from the index.html file in this directory C:\Program Files\Apache Software Foundation\Apache2.2\htdocs\ which is the default document root directory after Apache is first installed. The document root directory can be changed in Apache configuration file.

CLaaS®

# Apache Server Features

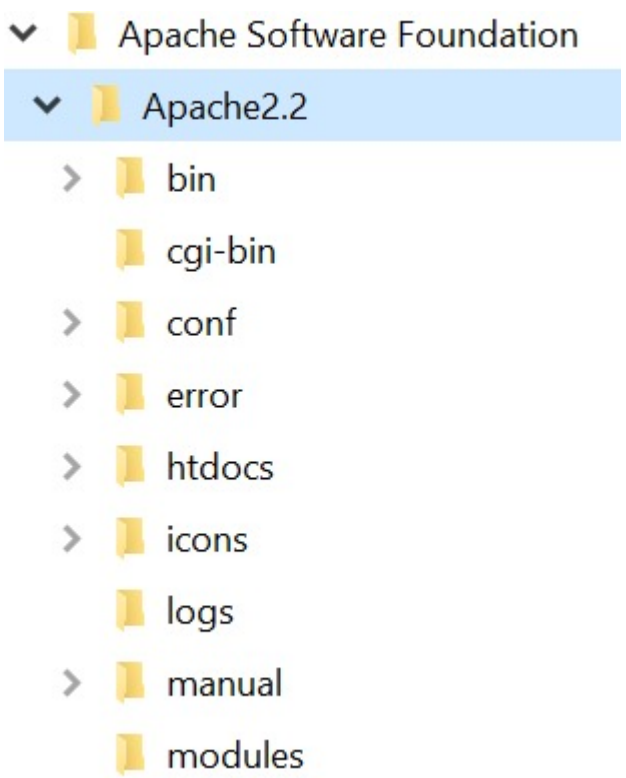
- **Server Features**
  - Apache Server can support multiple requests by spawning new thread to handle the incoming request
  - Apache serves content through 2 ports, port 80 for standard content and port 443 for Secured content
- **Server Side Programming Language Support**
  - Apache supports some common language interfaces which include Perl, Python, Tcl, and PHP.
  - Apache can run PHP / Python Natively or as CGI script
  - It also supports a variety of popular authentication modules like mod_auth, mod_access, mod_digest and many others.
- **IPv6 Support**
  - On systems where IPv6 is supported by the underlying Apache Portable Runtime library, Apache gets IPv6 listening sockets by default.
- **Virtual Hosting**
  - Apache will allow one installation instance to serve multiple websites. For instance one Apache installation can serve sse.afnog.org, ws.afnog.org etc

# Apache Server Features

- Simplified configuration (Using httpd.conf file)
- Apache is supported in almost all the major OS such as Windows, Linux, Unix, Mac etc.

# Apache Server Directory Structure

- After the installation is completed, Apache has directory structure in Windows as shown
- The httpd.conf in conf folder is the main configuration file for Apache
- htdocs is the document root for the default website
- logs folder contain all the error logs and access logs.



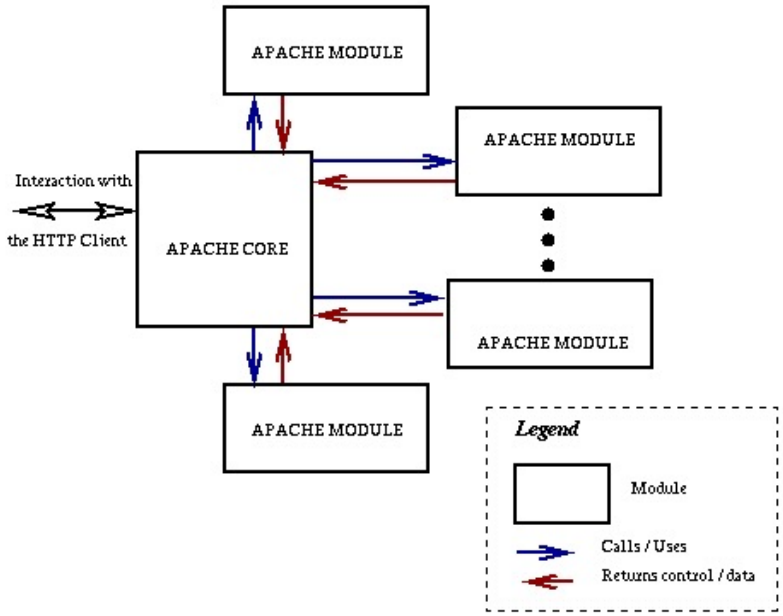

CLaaS®

# What is a Web Server ?

- Understands HTTP Protocol
- Generate appropriate responses
  - Clients “connect” to the machine
  - Clients send a “request”
  - Server reads request, generates “response”
  - Client interprets response appropriately
- Client asks for file
  - Server finds the file
  - Sends back a response header & file’s content
  - Server closes connection
- **Sample Web Servers**
  - Apache Web Server
  - IIS Web Server

# Introduction to Apache Server

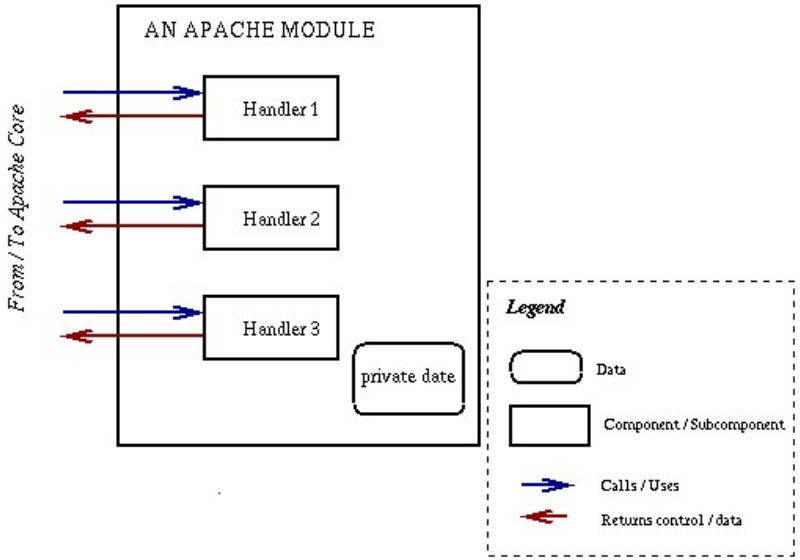
- Open source HTTP web server.
- Send & Receive HTTP responses.
- Maintained at Apache.org
- Building blocks
  - Apache Core
  - Apache Modules
- Easy to implement
- Easy to add & extend its abilities by different modules.




CLaaS®

# Overview of Modules

- Extend / overwrite and implement functionality to Apache.
- Do not know about other modules.
- Connected to the Apache core.
- Pass in formation to the core
- Core sends information to another appropriate module.
- Stable Apache Core

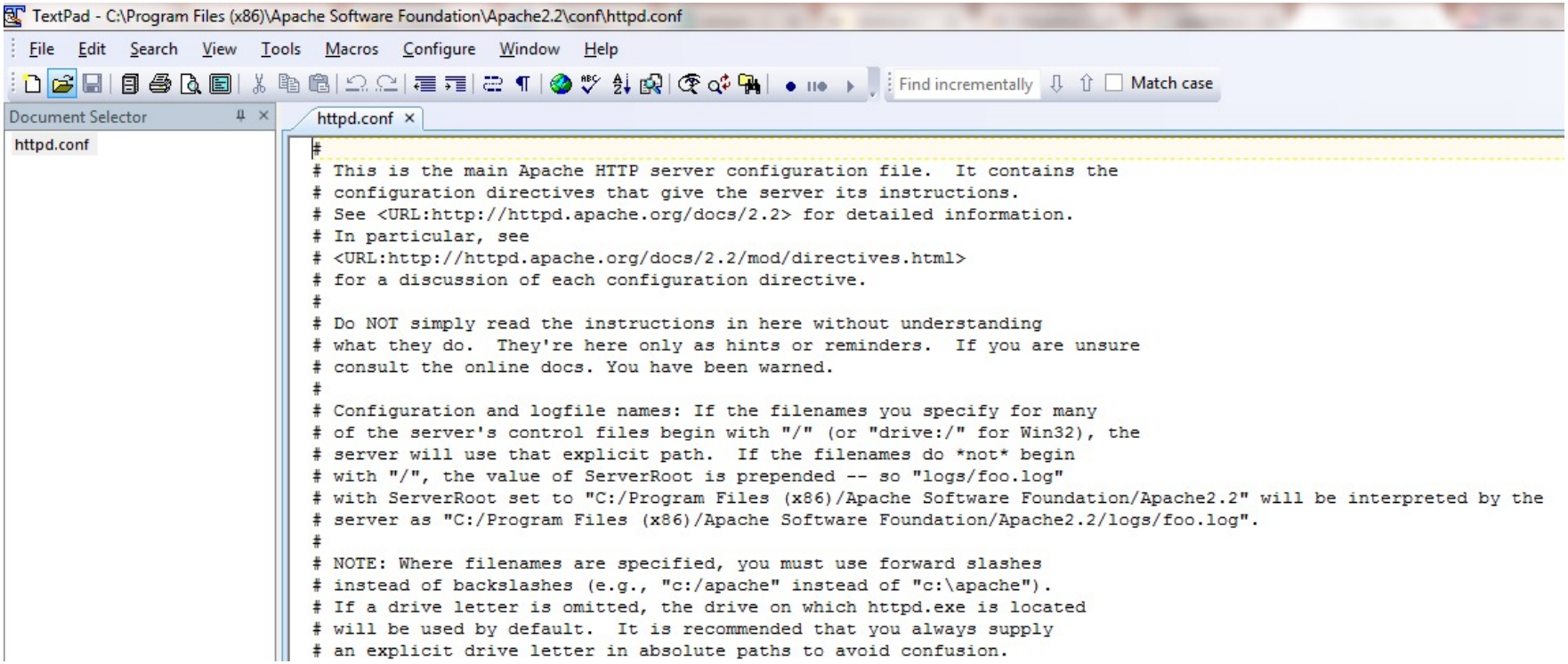



CLaaS®

# Main Configuration File(s)

- Main configuration file APACHEHOME/conf/httpd.conf.
- Large number of settings.
- **Backup**
  - Retain the copy of the original file.
  - Do NOT delete default lines, Comment them out.
- **Edit the httpd.conf configuration file**
  - Just a single line to create your server and get it running.

# Edit httpd.conf




- **ServerRoot**
  - Path to the server's configuration, error and log files.

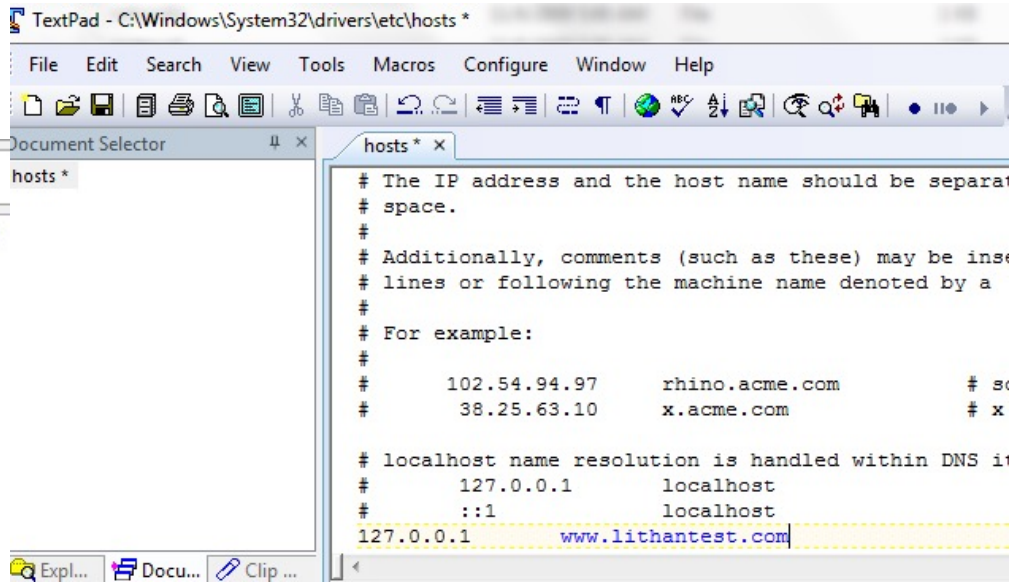
```
#
ServerRoot "C:/Program Files (x86)/Apache Software Foundation/Apache2.2"
```

CLaaS®

# Edit httpd.conf

- **ServerName**
  - Declare the name of your website.
  - Without registered DNS name, Use an IP address.
  - **Example** www.lithantest.com
  - **Add site to C:\Windows\System32\drivers\etc\hosts file**

    **127.0.0.1 www.lithantest.com**
- Visit www.lithantest.com


CLaaS®

# Edit httpd.conf

## DocumentRoot

- Where web documents (html files, images etc) files are located.
- The default directory is C:/Program Files (x86)/Apache Software Foundation/Apache2.2/htdocs.

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:/Program Files (x86)/Apache Software Foundation/Apache2.2/htdocs"
```

## Listen

- What ports to use for incoming connections. By default, port 80 is used.

```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
```

CLaaS®

# Advanced Setup httpd.conf

## ❑ Directory Tags

- Specify the configurations separately for each directory serving the web pages.
- Serve or deny content to specific IP ranges
- Limit access to certain files
- Set the behavior of pages

```
# Each directory to which Apache has access can be configured with respect
# to which services and features are allowed and/or disabled in that
# directory (and its subdirectories).
#
# First, we configure the "default" to be a very restrictive set of
# features.
#
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>
```

CLaaS®

# Virtual Hosts

- Run multiple websites from a server.
- IP-based or named-based
- Can use almost any option normally used in the **httpd.conf** file.
- Individual customized **httpd.conf** files nested inside the main httpd.conf file.

```
NameVirtualHost *:80
<VirtualHost *:80>
  DocumentRoot     c:/Temp/lt1
  ServerName       www.lithantest1.com
 <Directory "c:/Temp/lt1">
    Options All
    AllowOverride All
    Order allow,deny
    Allow from all
 </Directory>
</VirtualHost>
```

Declares multiple domains are hosted In a Single IP

Directory root from which To serve content

Defines domain that will be Used by the user

Configure permissions as needed

CLaaS®

# Hosting 2 Domain Names from a Single IP

- 2 domain names www.test1.com and www.test2.com
- 2 Folder **c:\temp\test1** and **c:\test\test2** to serve content
- Create 1 file each in **c:\temp\test1** and **c:\temp\test2** called **index.html** with different content
- Configure your hosts file in C:\Windows\System32\drivers\etc
  - Add 2 lines at bottom

```
127.0.0.1 www.test1.com
127.0.0.1 www.test2.com
```

  - No real domain or DNS configuration

CLaaS®

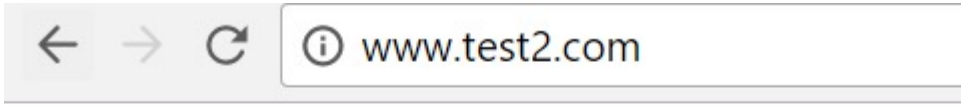
# Hosting 2 Domain Names from a Single IP

- Add below code in **httpd.conf** in the conf folder of Apache installation
- Backup **httpd.conf** before editing
- Add Below code at the end of the file (edit using notepad)

```
NameVirtualHost *:80
<VirtualHost *:80>
  DocumentRoot                c:/Temp/test1
  ServerName www.test1.com
 <Directory "c:/Temp/test1">
    Options All
    AllowOverride All
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
<VirtualHost *:80>
  DocumentRoot                c:/Temp/test2
  ServerName www.test2.com
 <Directory "c:/Temp/test2">
    Options All
    AllowOverride All
   Order allow,deny
   Allow from all
  </Directory>
</VirtualHost>
```

- Restart Apache after editing httpd.conf
- Below is the result accessing them from browser


test1




test2

# Web Logs

- find them out at **httpd.conf**
- **Error log**
  - Define **ErrorLog** directive.

```
#
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog "logs/error.log"
```

- **Access log**
  - Defined inside a virtual host

```
# The location and format of the access logfile (Common Logfile Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here.  Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#
CustomLog "logs/access.log" common
```

CLaaS®

# Reading the Logs

- An error log entry for a Python Script error would look something like:

[Tue Jan 03 11:04:46 2017] [error] [client 127.0.0.1] NameError: name 'cgi' is not defined\r

[Tue Jan 03 11:04:48 2017] [error] [client 127.0.0.1] Traceback (most recent call last):\r

[Tue Jan 03 11:04:48 2017] [error] [client 127.0.0.1] File "C:/Program Files (x86)/Apache Software Foundation/Apache2.2/htdocs/s7783549j/module1/iu14/PythonCrud.py", line 49, in <module>\r

[Tue Jan 03 11:04:48 2017] [error] [client 127.0.0.1] form = cgi.FieldStorage()\r

[Tue Jan 03 11:04:48 2017] [error] [client 127.0.0.1] NameError: name 'cgi' is not defined\r

- In this case the particular module cgi is undefined in the Python Script, that is why error is recorded.

CLaaS®

# Apache Modules

- Extensions that enhance the basic functionality of the Web server.
- **Module types**
  - Built-in modules
  - Loadable modules
- Type httpd –l

```
C:\Program Files (x86)
Compiled in modules:
  core.c
  mod_win32.c
  mpm_winnt.c
  http_core.c
  mod_so.c
```

- Type httpd –M

```
C:\Program Files (x86)\Apache S
httpd: Could not reliably deter
ing 192.168.1.102 for ServerNam
Loaded Modules:
 core_module (static)
 win32_module (static)
 mpm_winnt_module (static)
 http_module (static)
 so_module (static)
 actions_module (shared)
 alias_module (shared)
 asis_module (shared)
 auth_basic_module (shared)
 authn_default_module (shared)
 authn_file_module (shared)
 authz_default_module (shared)
 authz_groupfile_module (shared
 authz_host_module (shared)
 authz_user_module (shared)
 autoindex_module (shared)
 cgi_module (shared)
 dir_module (shared)
 env_module (shared)
 include_module (shared)
 isapi_module (shared)
 log_config_module (shared)
 mime_module (shared)
 negotiation_module (shared)
 setenvif_module (shared)
 php5_module (shared)
Syntax OK
```

# Apache Modules

- Open Command Prompt by running cmd
- Cd C:\Program Files (x86)\Apache Software Foundation\Apache2.2\bin
- Type httpd –l

```
C:\Program Files (x86)
Compiled in modules:
  core.c
  mod_win32.c
  mpm_winnt.c
  http_core.c
  mod_so.c
```

- Type httpd –M
- There is a wide range of modules available. We will review a number of more common ones. Please note that the list below is only partial and just briefly introduces the range of available modules.

```
C:\Program Files (x86)\Apache S
httpd: Could not reliably deter
ing 192.168.1.102 for ServerNam
Loaded Modules:
 core_module (static)
 win32_module (static)
 mpm_winnt_module (static)
 http_module (static)
 so_module (static)
 actions_module (shared)
 alias_module (shared)
 asis_module (shared)
 auth_basic_module (shared)
 authn_default_module (shared)
 authn_file_module (shared)
 authz_default_module (shared)
 authz_groupfile_module (shared
 authz_host_module (shared)
 authz_user_module (shared)
 autoindex_module (shared)
 cgi_module (shared)
 dir_module (shared)
 env_module (shared)
 include_module (shared)
 isapi_module (shared)
 log_config_module (shared)
 mime_module (shared)
 negotiation_module (shared)
 setenvif_module (shared)
 php5_module (shared)
Syntax OK
```

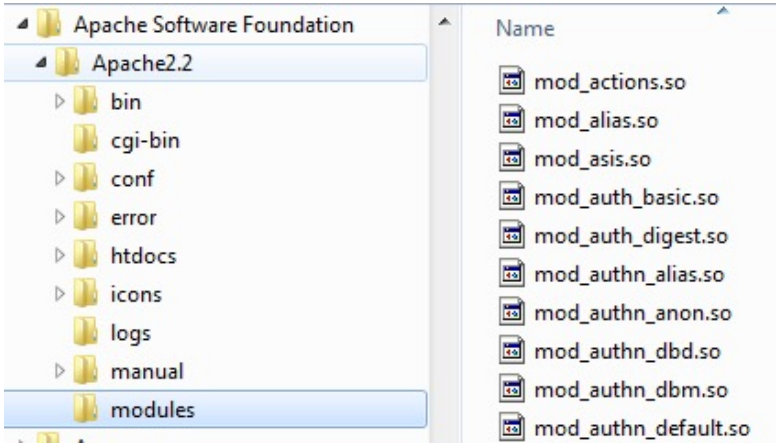
CLaaS®

# Apache Modules

## ❑ LoadModule

- Comment the line with #, if do not have to load a module.

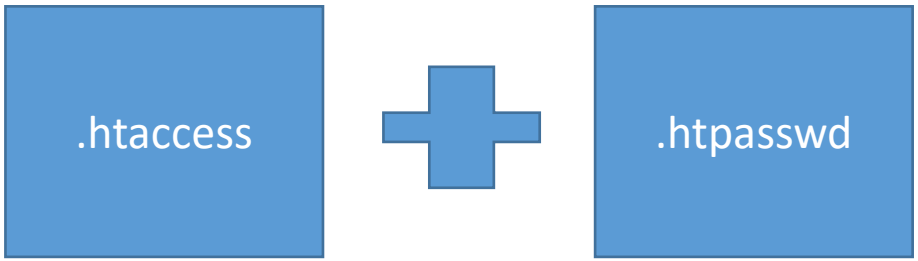
```
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_basic_module modules/mod_auth_basic.so
#LoadModule auth_digest_module modules/mod_auth_digest.so
#LoadModule authn_alias_module modules/mod_authn_alias.so
#LoadModule authn_anon_module modules/mod_authn_anon.so
#LoadModule authn_dbd_module modules/mod_authn_dbd.so
#LoadModule authn_dbm_module modules/mod_authn_dbm.so
```




CLaaS®

# Apache File Access Control

- Allows you to secure web pages in specific folders
- Secured folders will be prompted for User Name & Password
- To Setup this we need 2 files, They are
  - **.htaccess**
    - Files in the folder where it is present requires authentication
    - This file is created using notepad
  - **.htpasswd**
    - File has the user name and hashed password
    - Using the **htpasswd** tool which is in the <Apache installation folder>\bin to add or modify users




CLaaS®

# Apache File Access Control

- **.htaccess** file loaded every time a webpage is requested.

.htaccess × httpd.conf

```
AuthType Basic
AuthName "Restricted web page"
AuthUserFile "C:/Program Files (x86)/Apache Software Foundation/Apache2.2/htdocs/.htpasswd"
require valid-user
```

- **AuthType Basic**
  - Defines the type of authentication.
  - No encryption and the password hash is sent as clear text.
- **AuthName**
  - Popup window title string
- **AuthUserFile**
  - Defines the path to a file where user credentials are stored.
- **require**
  - Access files on successful authentication

CLaaS®

# Securing a Folder

- Create **.htaccess** file
- Create the **.htpasswd** file containing usernames & password.
- Place **.htaccess** in the directory which needs to be secured.
- Configure Apache to allow user authentication via .htaccess files.
- Restart the server.
- Test the results.
- **Note: Remember to change AllowOverride None to AllowOverride AuthConfig in httpd.conf & Restart Web Server**

```
#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   Options FileInfo AuthConfig Limit
#
AllowOverride AuthConfig
```

- **If you use virtual host remember to include this directive inside virtual host block**

CLaaS®

# Securing a Folder – Create .htaccess

- Create a folder to protect Ex: htdocs\ICNo.\module1\iu14.
- Create a file **.htaccess** in that folder with the following content

AuthType Basic

AuthName "Restricted web page"

AuthUserFile "C:/Program Files (x86)/Apache Software Foundation/Apache2.2/htdocs/.htpasswd"

require valid-user

# Securing a Folder – Create .htpasswd

- Create the file **.htpasswd**
- Open Command Prompt and **cd** to **C:\Program Files (x86)\Apache Software Foundation\Apache2.2\htdocs**
- Run the command
- ..\bin\htpasswd –c .htpasswd [yourname]
- Type in your password

```
C:\Program Files (x86)\Apache Software Foundation\Apache2.2\htdocs>..\bin\htpass
wd -c .htpasswd shrini
Automatically using MD5 format.
New password: *******
Re-type new password: *******
Adding password for user shrini
```

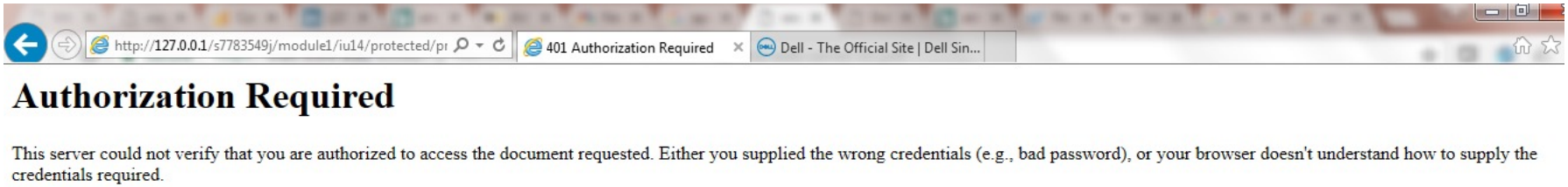
- The above procedure creates a file called .htpasswd under the C:\Program Files (x86)\Apache Software Foundation\Apache2.2\htdocs folder

CLaaS®

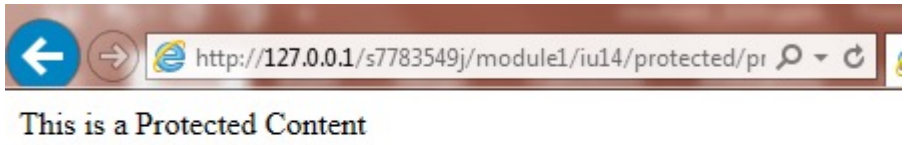
# Securing a Folder

- Now access http://127.0.0.1/[Your IC]/module1/iu14/protected/protected.html





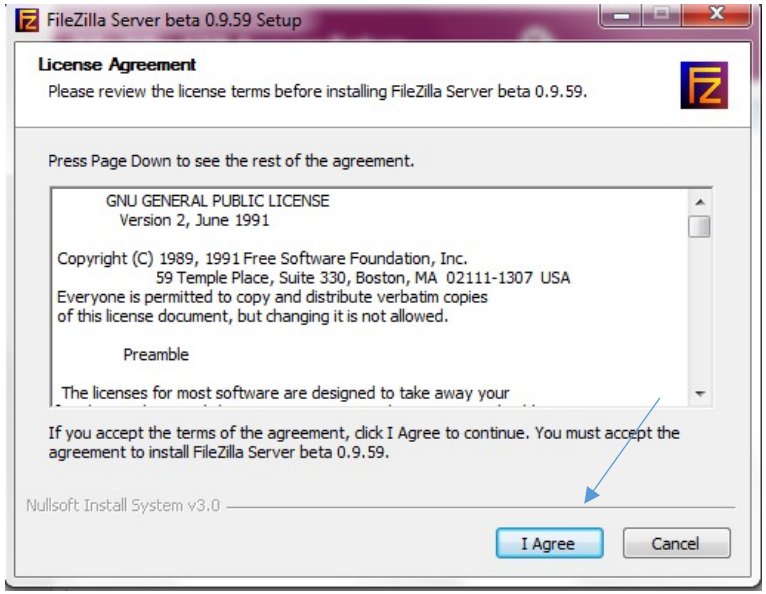

- Key in you user name and password

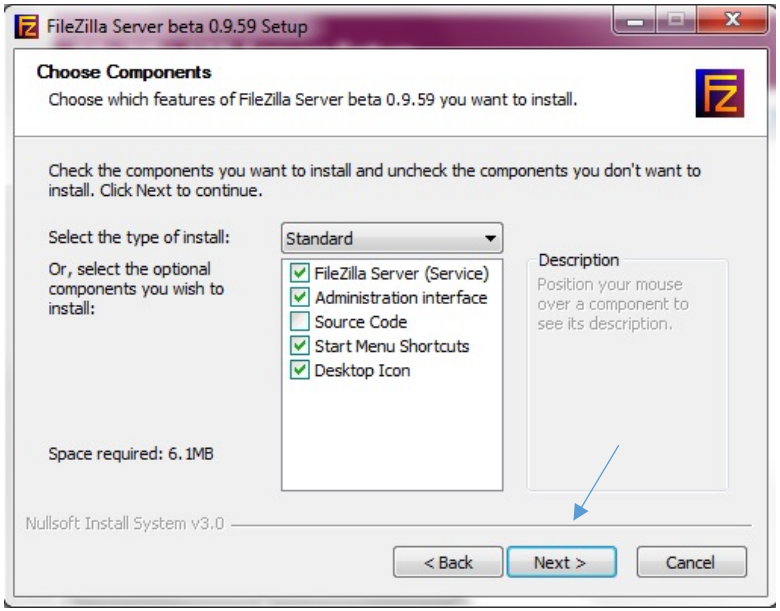



CLaaS®

# FileZilla FTP Server Setup

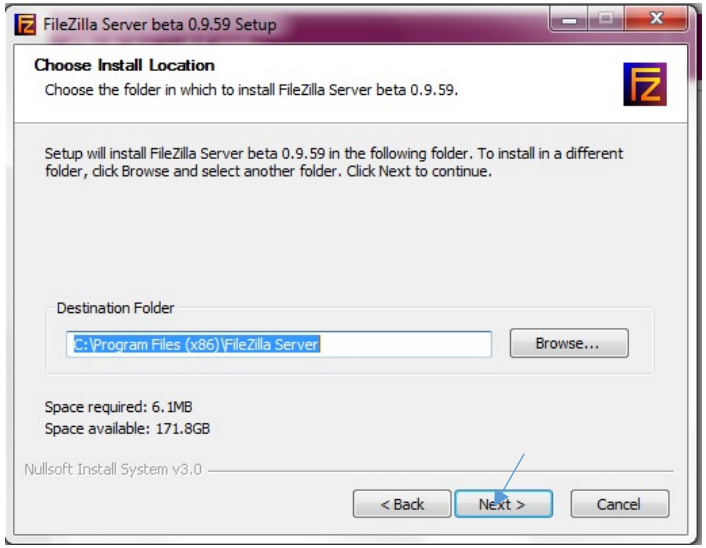
- ❑ Go to https://filezilla-project.org/download.php?type=server
- ❑ Download the file
- ❑ Run the Downloaded file

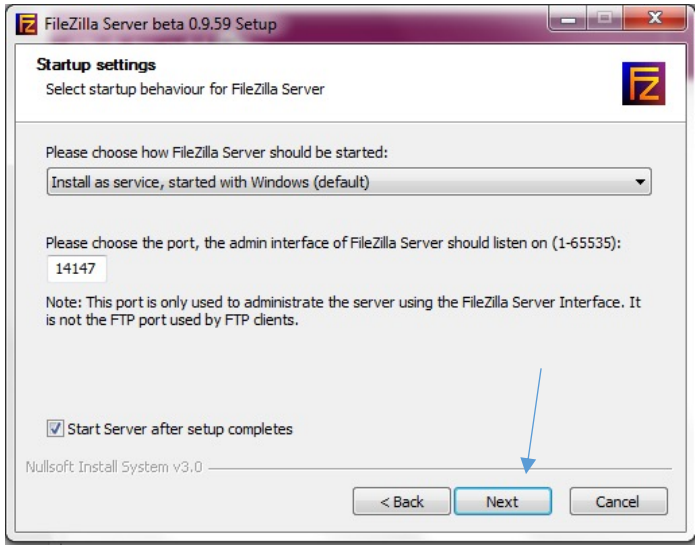


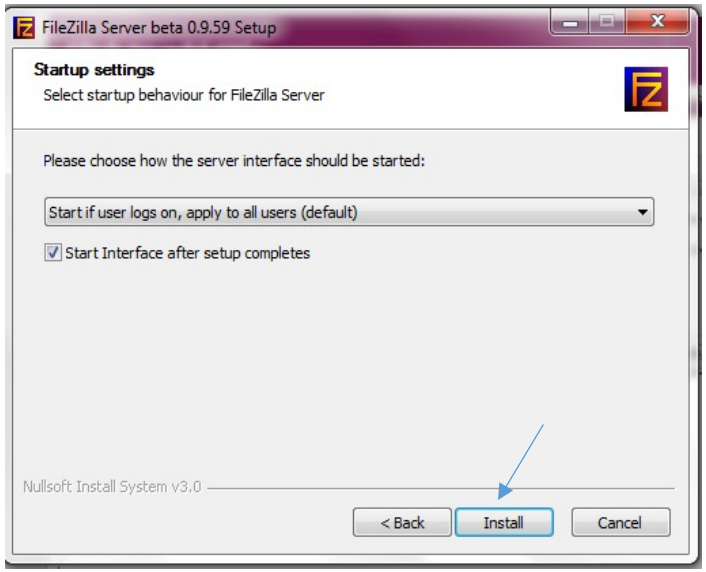



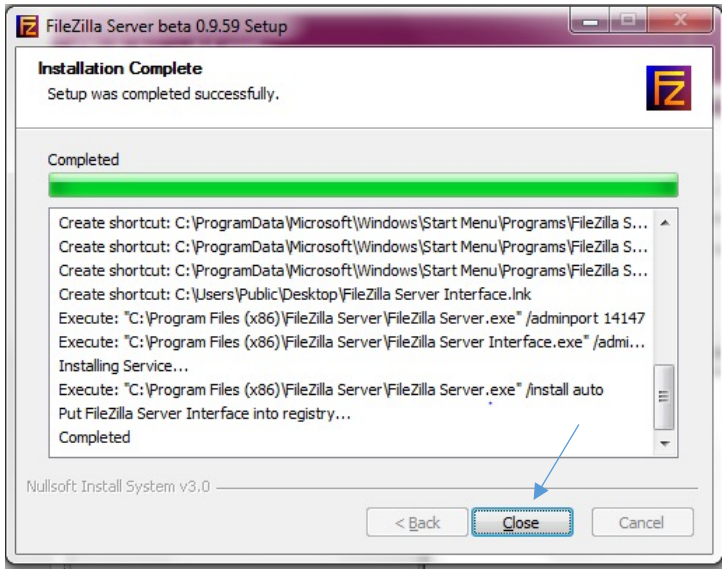

CLaaS®

# FileZilla FTP Server Setup



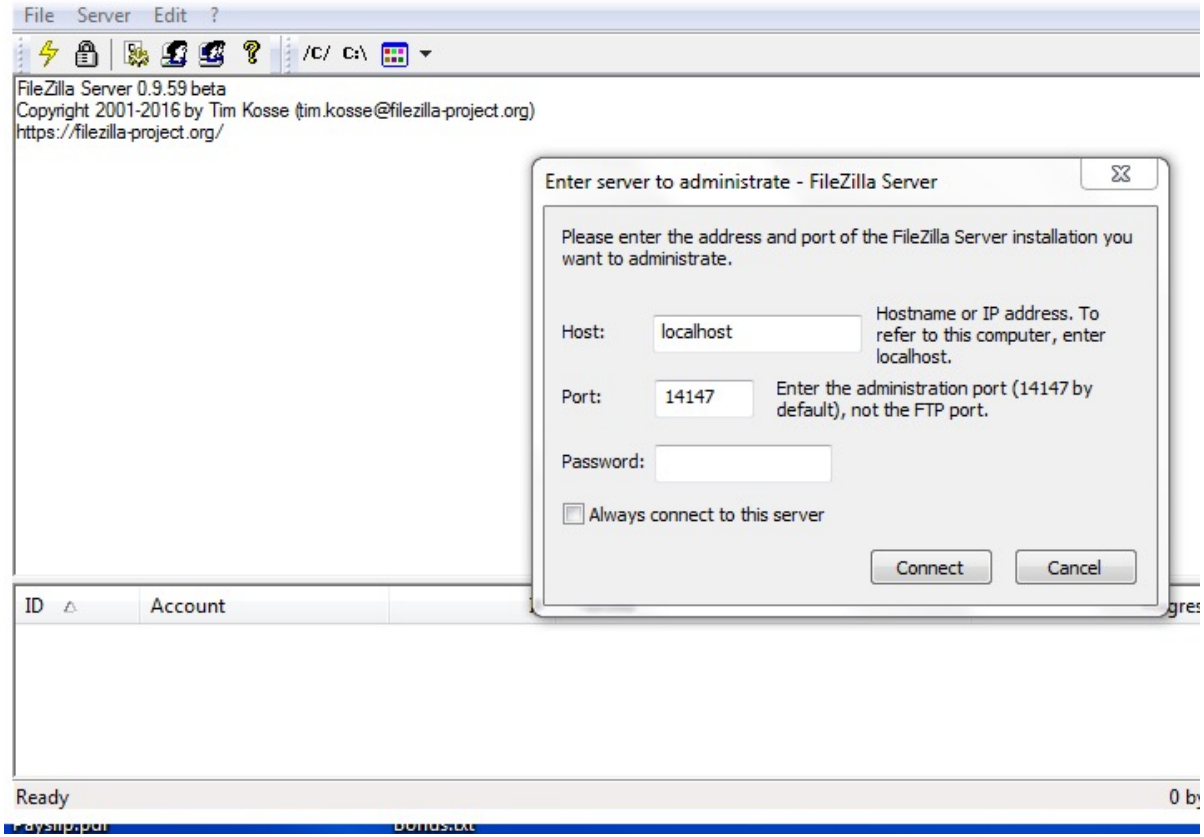




CLaaS®

# Configuring FileZilla

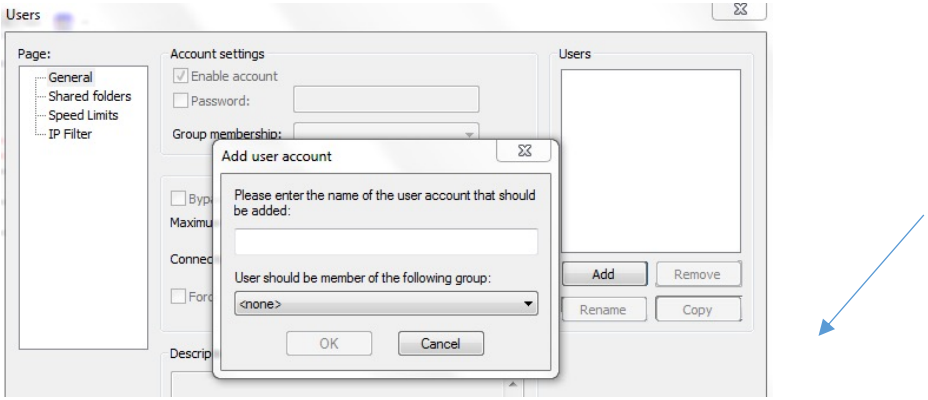
- Open FileZilla FTP Server Administration Interface
- Login By Clicking Connect



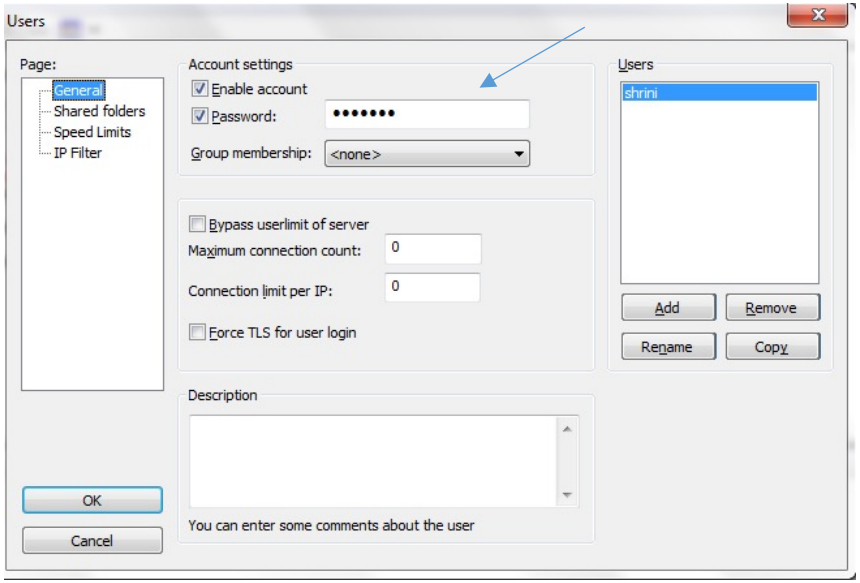

CLaaS®

# Configuring Users in FileZilla

- Lets Create Users by Choosing Edit -> Users in the Menu



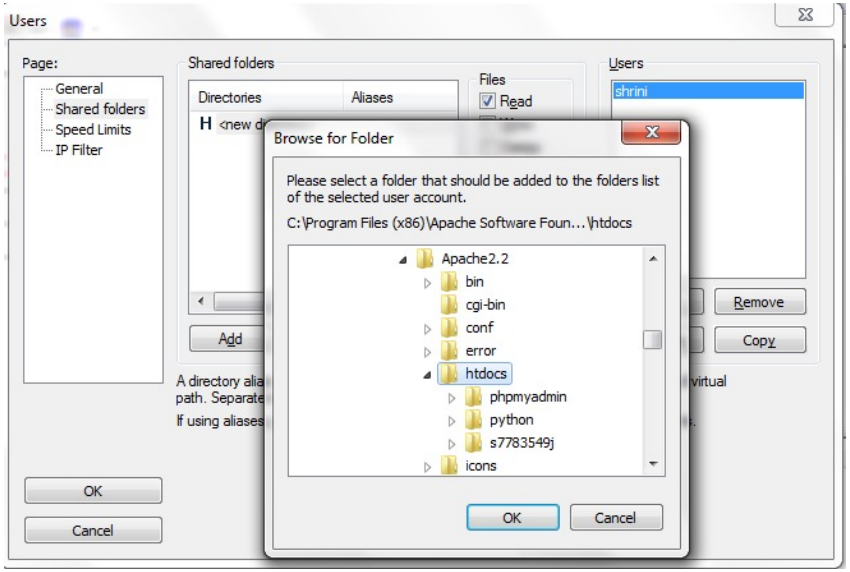

- Key in the User Name (Your name) & Click ok
- Set the Password by selecting password checkbox and key in password



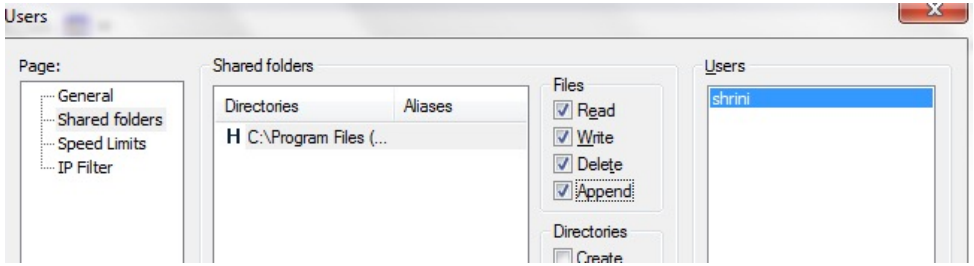

CLaaS®

# Share the Folder

- Click Shared Folders and Click Add Button and browse to C:\Program Files (x86)\Apache Software Foundation\Apache2.2\htdocs



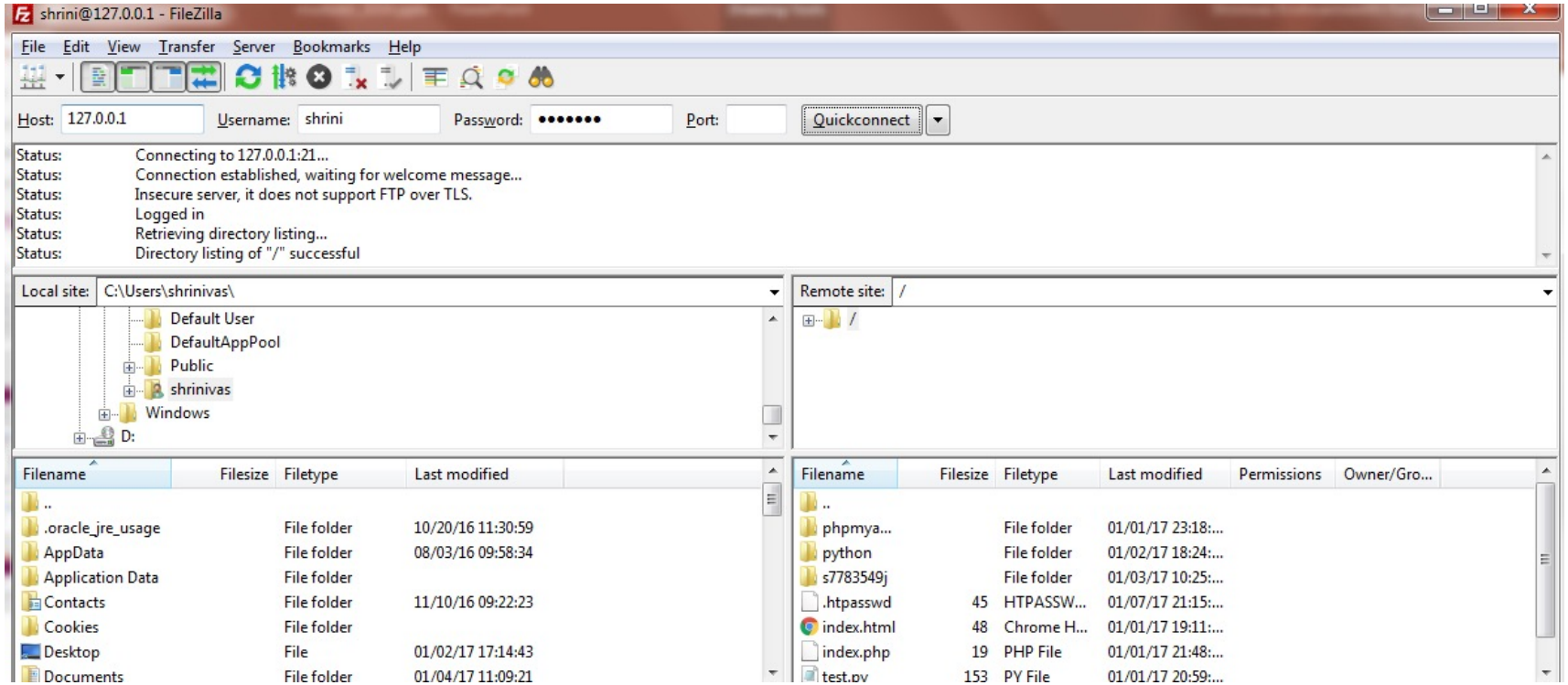

- After that Check Permissions Write, Delete, Append & Click Ok




CLaaS®

# Connecting to FTP Server

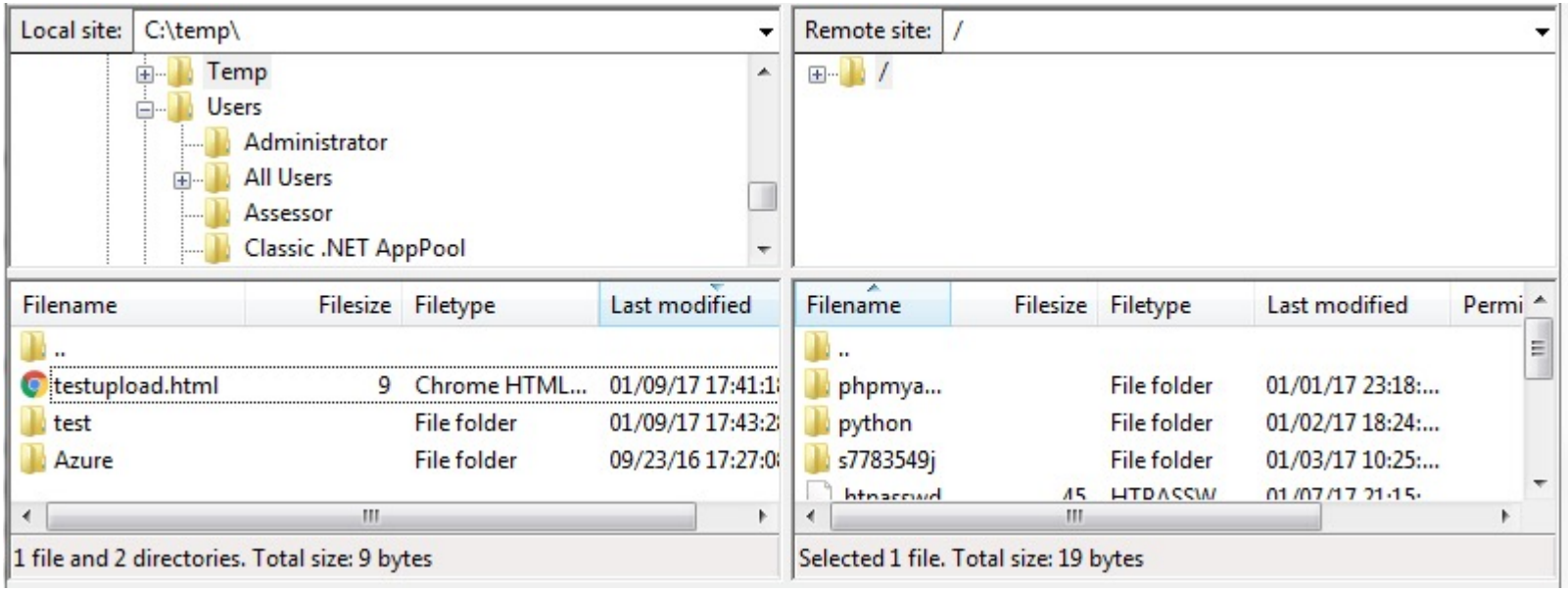
- Open FileZilla FTP Client
- Key in Host as 127.0.0.1 and User Name as the user name you setup, key in password you setup & click Quick Connect
- You will see the interface to upload & download files




CLaaS®

# Uploading & Downloading Files

- Create a file called testupload.html and c:\Temp
  - Put the text Test File in the file
- Drag & Drop from left to right to upload
- Drag & Drop from right to left to download file

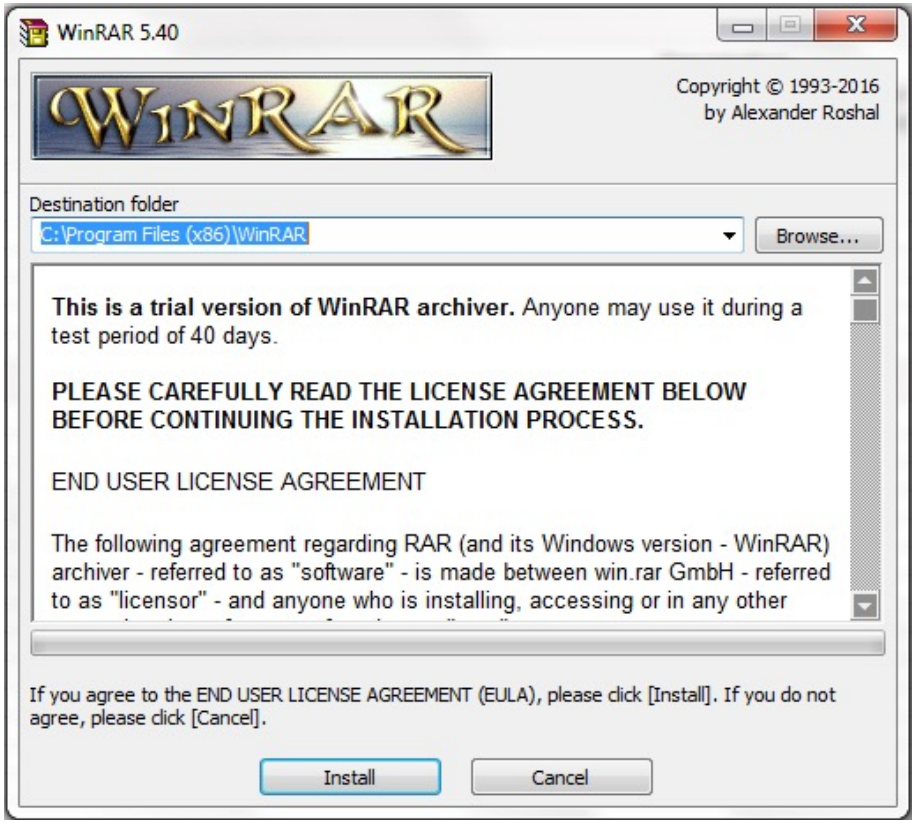




CLaaS®

# Files Backup & Restore Tools

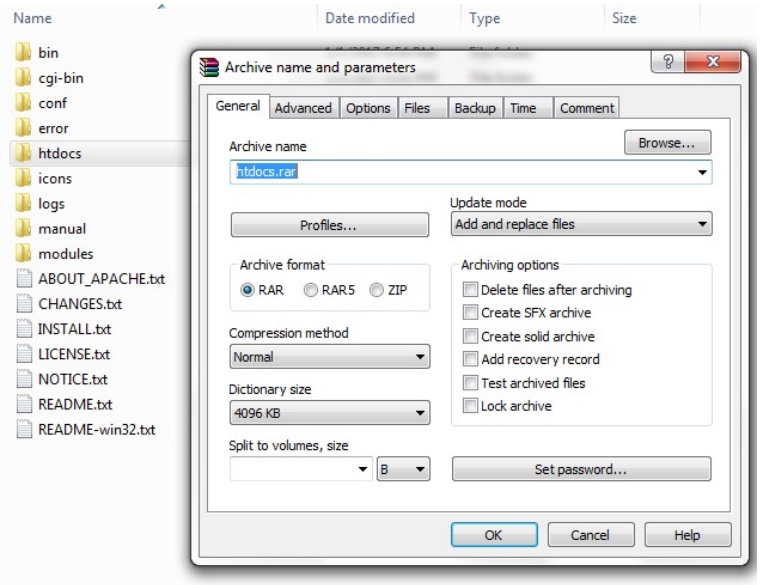
- If you do not have a Archive Tool, Download WinRar from http://www.rarlab.com/download.htm and Install it
- Run the Downloaded Exe and Click Install




CLaaS®

# Backup Files

- Open the folder C:\Program Files (x86)\Apache Software Foundation\Apache2.2 in Windows Explorer
- To Backup the htdocs folder and the httpd.conf file under conf folder
  - Create a Backup Folder Some where in File System
  - Right Click htdocs and click Add to Archive
  - Browse to Backup Folder Give the file name htdocs-DDMMYYYY.rar and click ok
- The Files will be zipped and placed in the Backup Folder




CLaaS®

# Restore Files

- Open Windows Explorer
- Go to Backup folder
- Right Click the RAR file
- Select Extract Here
- Then Move the files as needed to restore the files in htdocs folder
- Remember to backup httpd.conf file

CLaaS®

# Using Command Line for Archiving

- Compress a Folder in Command Line

rar a -r yourfiles.rar *.txt

- creates archive yourfiles.rar and compresses all .txt files in the current folder
- Give Appropriate folder where rar tool is located ex:

C:\winrar\rar.exe a -r yourfiles.rar c:\temp*.txt

# THANK YOU